Invert - инвертируем изображение

2-й курс/Закрытая зона

В графических редакторах есть операция инвертирования изображения (Photoshop: Image | Adjustments | Invert). Команда **Инвертировать** заменяет цвета и яркость точек текущего слоя на противоположные, получая негатив изображения. Тёмные области становятся светлыми, а светлые - тёмными.

Попробуем сделать это на Android. За основу возьмём [предыдущий пример](http://developer.alexanderklimov.ru/android/catshop/matrix.php#prescale) и удалим из него вторую кнопку.

Перейдем к коду. Каждый пиксель изображения содержит информацию о четырёх каналах: Alpha (прозрачность), Red (красный), Green (зелёный), Blue (синий). Значение прозрачности мы не трогаем, а остальные значения цветов вычисляем по формуле: 0xFF – CurrentValue.

Должен сразу предупредить, что операция очень ресурсоёмкая, ведь нам приходится пройтись по каждому пикселю изображения и поменять у него цвет. На больших картинках это занимает значительное время. Поэтому на всякий случай я добавил вызов всплывающего сообщения, пока вы его не увидите, не нажимайте на кнопку второй раз. Значит операция инвертирования еще продолжается. Операция инвертирования обратима. Если запустить инвертирование на уже инвертированном изображении, то получим обратно исходную картинку.

package ru.alexanderklimov.invert\_1098 ;

butInvert.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

BitmapDrawable mydrawable = (BitmapDrawable) imageView.getDrawable();

Bitmap b = mydrawable.getBitmap();

b = doInvert(b);

imageView.setImageBitmap(b);

Toast.makeText(getApplicationContext(),

"Инвертировано", Toast.LENGTH\_LONG).show();

}

});

public static Bitmap doInvert(Bitmap src) {

// Создадим новый bitmap с теми же настройками, как у источника

Bitmap bmOut = Bitmap.createBitmap(src.getWidth(), src.getHeight(), src.getConfig());

// информация о цветах

int A, R, G, B;

int pixelColor;

// размер изображения

int height = src.getHeight();

int width = src.getWidth();

// проходим через каждый пиксель

for (int y = 0; y < height; y++)

{

for (int x = 0; x < width; x++)

{

// работаем с отдельным пикселем

pixelColor = src.getPixel(x, y);

// сохраняем alpha-канал

A = Color.alpha(pixelColor);

// инвертируем байт каждого R/G/B канала

R = 255 - Color.red(pixelColor);

G = 255 - Color.green(pixelColor);

B = 255 - Color.blue(pixelColor);

// полученный результа записываем в этот же пиксель

bmOut.setPixel(x, y, Color.argb(A, R, G, B));

}

}

// Получаем новое изображение

return bmOut;

}

Запускаем проект и щёлкаем на кнопку, чтобы получить инвертированное изображение. Повторное нажатие вернёт изображение в исходное положение:

 

Р